

מהשבים, **שפות תיכנות ו-APL**

מאת ראובן אמיר



מדפסת/מקלדת יב"מ 3767

אחת ההתפתחויות בשטח המחשבים שהרחיבה בצורה ניכרת את הקף שימושם, היתה פיתוח שיטת הפעלה הנקראת "שתף-זמן" (Time Sharing). העבודה בשיטה זו נעשית באמצעות מסופים (עשרות או מאות) הקשורים טלפונית למחשב וכל משתמש פועל כאילו עמך המחשב רק לרשותו. עם התפשטות העבודה ב"שתף-זמן" הופיעו שפות-תיכנות מותאמות במיוחד לעבודה בשיטה זו. אחת מאלה היא שפת ה-APL ("A Programming Language"). שפה זו קלה לשימוש, מצטיינת בתמציתיות רבה ומתאימה למיגוון רחב של יישומים — החל בפיתרון בעיות מדעיות וכלה בהוראת מקצועות שונים בבתי-ספר תיכוניים.

על-אף הצרות הרבות שהביאה לעולם הקידומה הטכנולוגית, יש מקום לגאחה ולסיפוק כאשר אנו מצליחים לפענח עוד צופן של הטבע או לרתום עוד אחר מכוחותיו לשירות האנושות. בין הצעדים המהווים את ההתקדמות המדעית והטכנית יש גם כאלה שאין להם כל קשר לעולם הסובב אותנו. אין הם אלא פיתרונות, לבעיות שאנחנו בעצמנו יצרנו. במקרים אלה, נחא, מהשג האינטלקטואלי מרשים ככל-שהיה, מלחה הרגשת הסיפוק כצח-קוקיו של שדון ערמומי, הקורץ ומלחש שאין מה להתגאות כאן, פשוט לא היה צריך להמציא את הבעיה.

הנה, למשל, המחשב כפי שאנו מכירים אותו כיום, עם כל המיטען הכבד של הכלים לאירגון-מידע שנוצרו סביבו, הוא פרי רוחו של האדם; ללא-ספק, השג אינטלקטואלי כביר. אך, הסתבר, בשלב די מוקדם של פיתוח המחשבים, שהעלחתנו המסחררת הביאה, ככנפיה גם בעיה חמורה: זו של התקשורת עם יציר-כפינו. כידוע, כל מה שמתרחש במחשב המסובך ביותר הוא "פשוט" זרמים אדירים של מיליארדי סיביות (ספרות בינ-אריות), דהיינו, מעבים של "אחד" או "אפס", המפוקדים ע"י סיביות נוספות, נעים בסדר ומיטטר מפליאים ומביאים, לבסוף לתוצאה הרצויה לנו. בשלבי הפיתוח הראשונים של המחשבים לא היתה כל דרך לשלוח את הסיביות הללו לדרכן אלא באמצעות סיביות שהכניס האדם למחשב. עדין זכורים לרבים הימים בהם החזן המחשב

הראשון שניבנה במכון ויצמן בסרט-נייר ארוך, שבו סימל נקב במקום מסוים את הסיפורה 1 והעדר נקב את הסיפורה 0; כל בקשה שהופנתה אל המ-חשב חייבת היתה לעבור קודם תרגום לסידרה של ספרות בינאריות, שהונה לתוכו באמצעות סרט כזה. בשל הצורך בתרגום הנתונים לשפת-המחשב היה השימוש מסורבל ואיטי. לפיכך הוש-קעו מחשבה ומרץ רבים, לפישוט ולקיצור התה-ליך. המחשב עצמו נרתם לתפקיד של תרגום הוראותיו לזרם סיביות, וכך נולד הרעיון של "שפת תיכנות", דהיינו, שפה המובנת הן לאדם מן-השורה והן למחשב.

במשך השנים נכתבו שפות-תיכנות רבות, למט-רות שונות, כמו אסמבלר (Assembler), שהיא שפה הקרובה במיבנה שלה למיבנה שפת-הסיביות שהוזכרה לעיל; COBOL — שפה לעיבוד-נתונים מסחרי; FORTRAN — שפה הנוחה לביצוע חי-שובים הנדסיים ומדעיים; PL/I — שפה טובה למד-ענים ולמעבדי-נתונים גם יחד; LISP — שפה לבי-צוע פעולות באלמנטים סימבוליים; וכן שפות אחר-רות רונמת ALGOL, SNOBL ועוד כהנה וכו-וכהנה. השיפור ביעילות התקשורת עם המחשב עקב הכנסתו לשימוש של שפות אלה היה מהפכני. שפת ה-FORTRAN למשל פתחה את הדרך לניצול-בקנה מידה נרחב של המחשב ע"י המה-רסים למיניהם, הסטאטיסטיקאים, המתמטיקאים-השימושיים ואחרים. השימוש בשפות החדשות לא היה נטול בעיות ומיגבלות, אך ההלם הראשוני,

ראובן אמיר הוא בוגר הטכניון בחיפה שבו קיבל תואר B.Sc. (1956) ו-M.Sc. (1960) בהנדסת-מים. עבד בחברת תיכנות-המים לישראל, בה ניהל את המחלקה לסטאטיסטיקה ומתימטיקה שימושית. ב-1967 קיבל תואר דוקטור באוניברסיטת סטנ-פורד, קאליפורניה; נושא עבודתו היה יישום חקר-ביצועים להנדסת משאבי-מים. עתה הוא עובד ב-י.ב.מ. ישראל, עוסק בשתף-זמן ובשיטות קלט-נתנים.

שבא עם השימוש בערצמה הגלומה במחשבים, מילא את המשתמשים בהערצה, שחיפתה על הפגמים השונים שבשפות אלה.

כיצד מתגברים על פגמים?

ה"ארכיטקטורה" של מחשבים הכתיבה דפוסיים מסוימים לגבי הצעדים שיש לנקוט כדי לבצע את השלבים הדרושים לקבלת התוצאה המבוקשת, כלומר — מה שקרוי עתה "תוכנית-מחשב". על-אף העובדה שמפתחי השפות השונות הציבו לעצמם כמטרה את הפחתת המאמץ התיכנותי הדרוש, אין רוב השפות נקיות מתוספות והגבלות, שהן מלאכותיות לחלוטין מנקודת ראותו של המשתמש. אך הן משקפות את צורת-הפעולה המשתמשת של מחשב. השימוש בשפות רבות אף כרוך בקיום כללים שרירותיים, שמטרתם להקל על פעולת התרגום משפת-התיכנות לשפת-המכונה ולקצר את זמן הביצוע של התוכנית. לדוגמה, יש שפות שבהן חייב המתכנת לציין מראש האם מישתנה מסוים מייצג מיספר שלם או שבר עשרוני; וכן, כאשר רוצה המתכנת לבנות טבלות שונות, הוא חייב, כמעט ללא יוצא-מן-הכלל, לקבוע בראש התוכנית את ממדי הטבלה, למרות שגורל זה בעצם איננו ידוע מראש. לא-פעם מח-טיא המתכנת בניהוש, ונתוניו גולשים מן הטבלה, או שנשארה בה מקום בלתי-מנוצל.

כל-עוד היתה ידיעת התיכנות בעיקר נחלת מקצוענים, שהתייחסו לכללים מן הסוג שמנינו לעיל כאל סודות מקצועיים, לא היה לחץ חזק לשינויים. אולם, לפני שנים אחדות התחוללה מהפכה ששינתה את פני הדברים תכלית שינוי. הכחנה למיוג שחל בין שתי טכנולוגיות אשר הטביעו את חותמן על הדור: מחשבים ותקשורת. אחד מצאצאי מיוג זה היא שיטת-עבודה במחשב הנקראת שיתוף-זמן (Time Sharing). להלן שז"מ (T/S). מחשב שעליו מופעל שז"מ מאפשר למשתמשים רבים, היכולים להימצא בכל מקום שממנו אפשר ל"טלפן" למחשב, להתקשר אליו דרך קו-טלפון רגילים ולשוחח איתו באמצעות מסופים (terminals). המסופים הם מכונות-כתיבה או מסכי-טלחיה עם מקלדת (keyboard) הקשורים טלפונית למחשב, וכל משתמש חי בהרגשה שמסופו הוא מחשב פרטי שלו לכל-דבר. שז"מ התפשט במהירות. תחילה באוניברסיטות ואח"כ בשירות לציבור הרחב, כמעט בכל מדינות העולם, וישראל בכלל-זה. אצלנו נמצאים עדיין מרבית המשתמשים בשז"מ בין כותלי האוניברסיטות. עקב הגפתחות זו, גדל תורן זמן קצר מיספר המשתמשים במחשב כמה מונים; ובין המשתמשים החדשים היו עתה רבים שהמחשב איננו עיסוקם העיקרי אלא מכשיר-עזר, לפיתרון בעיותיהם בתחומים רבים ומגוונים. אלה התייחסו בהרבה פחות סובלנות לספיימים המלאכותיים בשפות השונות. יש לזכור שבעת עבודה ב-שז"מ החיטכון במיספר ההקשות הדרוש להחדרת רעיון מסוים למוחו של המחשב הוא חיוני: ראשית, רוב המשתמשים אינם כתבניות מקצועיות, ושנית, זמן הוא כסף.

על רקע זה נוצרה שפה חדשה, השונה בתפישתה לחלוטין מכל שפות-התיכנות הקודמות, נקיה מכל סוג של תקורה (overhead) המסרבלת שפות אחרות (פקודות כגון, dimension, begin, declare, end, stop, format), קלה ללימוד ועם-זאת מתאימה ליישומים מתוחכמים ביותר. העיקרון שהינחה את ממצאה היה שלא להטיל כל מיגב-

לות וכללים שרירותיים על המשתמש. המדובר בשפת ה-APL (ראשי תיבות של A Program, ming Language), אשר ראתה את אור העולם כתחת הדקטור של ק"א אייוורסון (Iverson). באוניברסיטת הארחארד, ארה"ב, בשנת 1962. מטרתו העיקרית של אייוורסון בעת פיתוח השפה היתה לא-דחקה תקשורת עם מחשבים אלא יצירת כלי שבעזרתו אפשר להגדיר בצורה מדויקת וחד-משמעית תהליכים לעיבוד מידע ("אלגוריתם-מים"). החל מהפשוטים-יחסית, כמו פיתרון של משוואה ריבועית או מציאת מחלק משותף גדול ביותר, וכלה בתהליכים חשוביים מורכבים ביותר. עם כל תכונותיה הטובות, אפשר להניח שיצירה זו לא היתה זוכה לפירסומה הרחב לולא הפכוה אייוורסון ועמיתיו, במרכז-המחקר של חברת י.ב.מ. ביורקטאון הייטס, לשפת-תיכנות למחשבים. זמן קצר לאחר גיבוש גרסתה הראשונה.

השפה קנתה לה חסידים נרחבי-העולם והאימרה הנפוצה היא שאת APL אין לומדים, ב-APL נדבקים! ומעניין שעתה, כאשר למעשה כל המשתמשים ב-APL הכירו שפה זו באמצעות המחשב, נסגר המעגל חזרה לרעיונו הראשון של אייוורסון והולכת ומשתרשת הדיעה ש-APL יכולה לשמש ככלי-תקשורת מצויין לא-רק בין אדם למחשב אלא גם בין אדם לאדם: להגדרת מערכות תוכנה (software), להגדרת אופן פעולתו של מחשב וכן בחינוך, כאמצעי להקניית הרגלי-חשיבה ובי-טוי מדויקים.

APL על רגל אחת

על הקורא לתאר לעצמו שהוא יושב ליד המ-סוף דמוי מכונת-הכתיבה, מחייג אל המחשב בטלפון ומתקתק על המסוף את הצופן, דהיינו: מיספר המשתמש שלו.

כיצד מתנהלת "שיחה" בין המסוף והמחשב? השאלה המופנית למחשב מורפסת על המסוף מימין לשמאל, תשובת המחשב ניתנת סמוך לשו-ליים, משמאל.

להלן כמה דוגמות פשוטות:

(1) המשתמש מרפס: $12 + 2 \times 12$ והמחשב עונה: 36

(2) ואם נתקתק: $3 \times 4 + 5$ נקבל כתשובה: 17

לא, אין כאן טעות בחשבון; אין הירארכיה של פעולות חשבון ב-APL, והפעולות מתבצעות לפי הסדר, "מימין לשמאל".

אך אם נרפס: $(3 \times 4) + 5$ נקבל: 17

הואיל והוספנו סוגריים, שפירושם — המכפלה קודמת.

בכל אחת מן הדוגמות שלעיל חישובנו את ערכו של ביטוי בודד; באותה קלות ניתן גם לחשב, כביכול בעת ובעונה אחת, את הערכים של ביטויים אחרים. לדוגמה, נניח שאנו רוצים לחשב "מכנה אחת" 5^3 , 12^2 ו- 2^4 ; נעשה זאת ע"י כך שנרפס במסוף:

(4) $2 \quad 12 \quad 5 \quad * \quad 4 \quad 2 \quad 3$ והתשובה תהיה: 16 144 125

הכוכבית מסמנת העלאה בחזקה; משמאלה רשו-מים המספרים 2, 12, ו-5. שאותם יש להעלות בחזקה, ואילו מימין לכוכבית נכתבו המעריכים המתאימים.

קבוצות מיספרים כמו 5, 12, 2 או 3, 2, 4 נקראות בענה המתימטית "קטורים" והמיספרים המהחים את הקבוצה נקראים "רכיבים" (אלמנטים). כפי שראינו בדוגמות דלעיל, ההפרדה בין רכיבי קטור נעשית באמצעות רווח

הפעולה שבוצעה לעיל היא דוגמה פשוטה ליכולתה של APL לבצע פעולות במסביל, צורה פרימיטיבית זו של מקבילית קיימת גם בשפות-מחשב אחרות כמו PL/I ו-BASIC לשם הש-חאה, להלן תכנית בשפת פורטראן המבצעת את הפעולות בדוגמה (4) עם הסברים לפקודות המ-החת את התוכנית.

הקצאת מקום בזיכרון של המחשב ל-3 קטורים בני 3 רכיבים כל אחד: DIMENSION K(3), L(3), M(3)

פקודה למחשב לקרוא את הערכים של הקטורים K ו-L: READ (1,5) K,L

פקודה המציינת שערכי הרכיבים שיקראו יופיעו בשדות באורך 8 תחיים כל אחד: FORMAT (618)

הוראה לבצע 3 פעמים את פקודה מס' 2: DO 2 I = 1,3

העלאה של K(I) בחזקת L(I) ואיחסון התוצאה בשם M(I):

$2 \quad M(I) = K(I) * L(I)$

פקודה להדפסת התוצאות: WRITE (1,6) M

פקודות המציינות את סיום התוכנית: STOP END

נחזור ל-APL. קטור המכיל רכיבים אחרים ניתן לציין בסימן יחיד, הנקרא מישתנה. למשל:

(5) $A \leftarrow 3 \quad 4 \quad 5$

את הערך 3 4 5 ייצג מעתה המשתנה A. המערכת רשמה לפניה אוטומטית שהמישתנה A בא במ-קום קטור של שלושה רכיבים מיספריים. המסוף אינו משיב דבר הואיל ולא נתבקש לכך. גם כמו בשפות-תיכנות אחרות, ניתן גם ב-APL לבצע פעולות לא רק על מיספרים אלא גם על אותיות:

(6) $A \leftarrow \text{'DUCKPOND'}$

המשתנה A מייצג עתה קטור של אותיות בעל 8 רכיבים בשום מקום אין צורך לציין מראש את אורך הקטור או את סוג הרכיבים שבו. להיפך, קיימת פעולה המסומנת באות היחידית רו (ρ) שבאמצעותה אפשר להיחדע מה מיספר הרכי-בים בקטור A:

לשאלה: ρA תתקבל התשובה: 8

בקשה להדפסת ערכו של מישתנה מתבצעת תוך הדפסת שמו:

אם נרפס במסוף: A תתקבל במקרה התשובה: DUCKPOND

ב-APL, פעולות החיבור, החיסור, הכפל והחצי-
לוק מסומנות כפי שמקובל באלגברה; פעולות
אחרות מסומנות בסימנים מיוחדים או באותיות
יחידות קטנות. כאשר הסימנים מסמלים פעולה
המתבצעת על מישתנה אחד בלבד, נרשם המיש-
תנה תמיד מימין לסימן המציין את הפעולה.
כאשר הפעולה מתבצעת על שני מישתנים, הם
נרשמים משני צידי הסימן.

- (9) למשל, 5 עצרת $5!$ יכתב כך: 5!
תשובת המחשב: 120
או המקדם הבינומי $\binom{5}{3}$
(10) יכתב כך: 5! 3
תשובת המחשב: 10
יצירת וקטור של מיספרים
(11) עוקבים יסומן כך: 4 2
תשובת המחשב: 1 2 3 4

סוג אחר של פעולות הן אלה הקרויות "לו-
גיות" — דהיינו כאלה שהתוצאה שלהן "אמת"
או "שקר"; אמת מסמנים ב-1 ואילו שקר ב-0.
להלן נראה פעולות כאלה הן ברכיבים בודדים
(סקאלרים) והן בהקטורים.

- (12) $3 < 4$
1
(13) $3 \vee 4$
0
(14) 'DODO' \neq 'DODA'
0 0 0 1

במקרה זה התקבלה התוצאה בעזרת השחאה בין
כל רכיב של ההקטור DODO לרכיב המתאים של
ההקטור DODA.

צירוף וקטורים נעשה באמצעות הפעולה שסי-
מנה פסיק (,):

- $A \leftarrow 2 \ 3$
 $B \leftarrow 4 \ 5$
(15) A, B
2 3 4 5

פעולות כאלה ניתן לבצע גם בין סקאלר להקטור
כמו בדוגמות שלהלן:

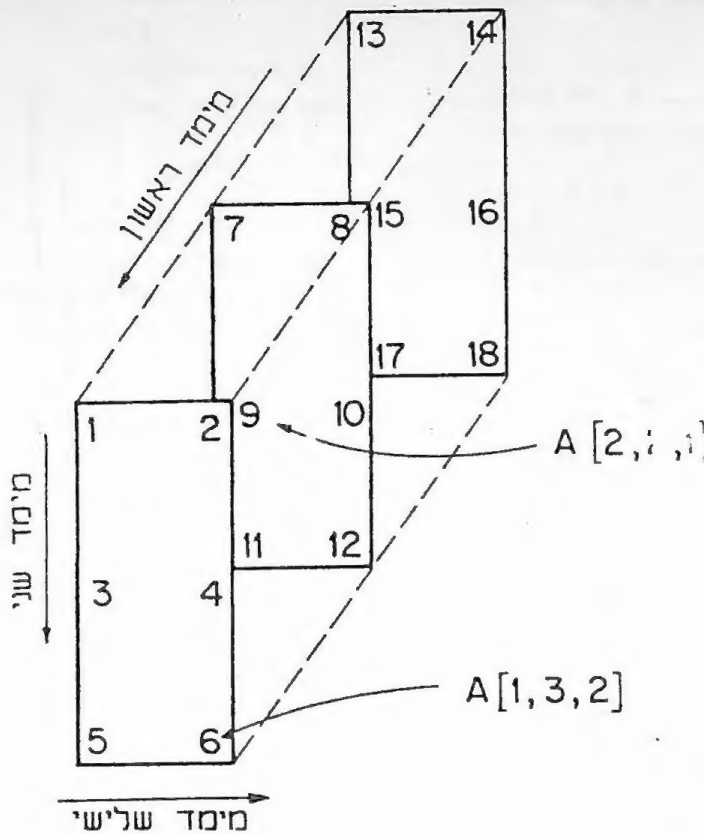
- (16) $2 + 1 \ 3 \ 4$
3 5 6
או, לדוגמה, אם נתקתק על המסוף:
(17) 2×4

נקבל כתשובה:
זאת משום ש-4 נותן את ההקטור 1 2 3 4
וכל אחר מרכיביו של וקטור זה הוכפל פי 2.
לעומת זאת, $4+2$ ייתן את ההקטור
2 3 4 5 6, מאחר ו-1 מחשב את הביטוי
מימין לשמאל; הסימן $\{$ מתייחס לא רק ל-4 אלא
ל- $4+2$, ומה שמתבצע למעשה הוא 6×2 ולכן
מתקבלת התוצאה שלעיל.

¹ עצרת פירושה כפל של מיספרים עוקבים החל
מ-1 וגמור במיספר המצוין. כלומר, "5 עצרת"
פירושו $1 \times 2 \times 3 \times 4 \times 5$.

² ביטוי היא ביטוי אלגברי המורכב מסכום או
הבדל שני אברים. את המקדם הבינומי מחש-
בים באמצעות נוסחה. במקרה שלנו יתחשב
המקדם הבינומי כך:

$$\binom{5}{3} = \frac{5!}{3!(5-3)!} = \frac{1 \times 2 \times 3 \times 4 \times 5}{1 \times 2 \times 3 \times 1 \times 2} = \frac{5 \times 4}{1 \times 2}$$



ציור 1: ציון מקומו של רכיב בתוך טבלה תלת-ממדית (ראה טכסט).

מאל לימין. לדוגמה, $A[1; 3; 2]$ יסמן את
הרכיב הנמצא ב"שיכבה" הראשונה, בשורה הש-
לישית, ובעמודה השנייה. על זוגות של טבלות
כאלה אפשר לבצע את כל הפעולות שאפשר
לבצע על שני סקאלרים (מיספרים בודדים),
והפעולה מתבצעת במקביל על כל הזוגות של הר-
יבים המתאימים משתי הטבלות.

נוסף לפעולות שלעיל, ה"סקאלריות" ישנן
עוד פעולות שכולן מעורבים וקטורים או זבלות,
והן התורמות העיקריות לצביון המיוחד של APL.

עד עתה ראינו שהמישתנים ב-APL יכולים
לייצג וקטורים וסקאלרים; המישתנים יכולים
גם לייצג טבלות בעלות מיספר ממדים כלשהו,
כמו הטב τ תלת-ממדית בציור מס' 1.

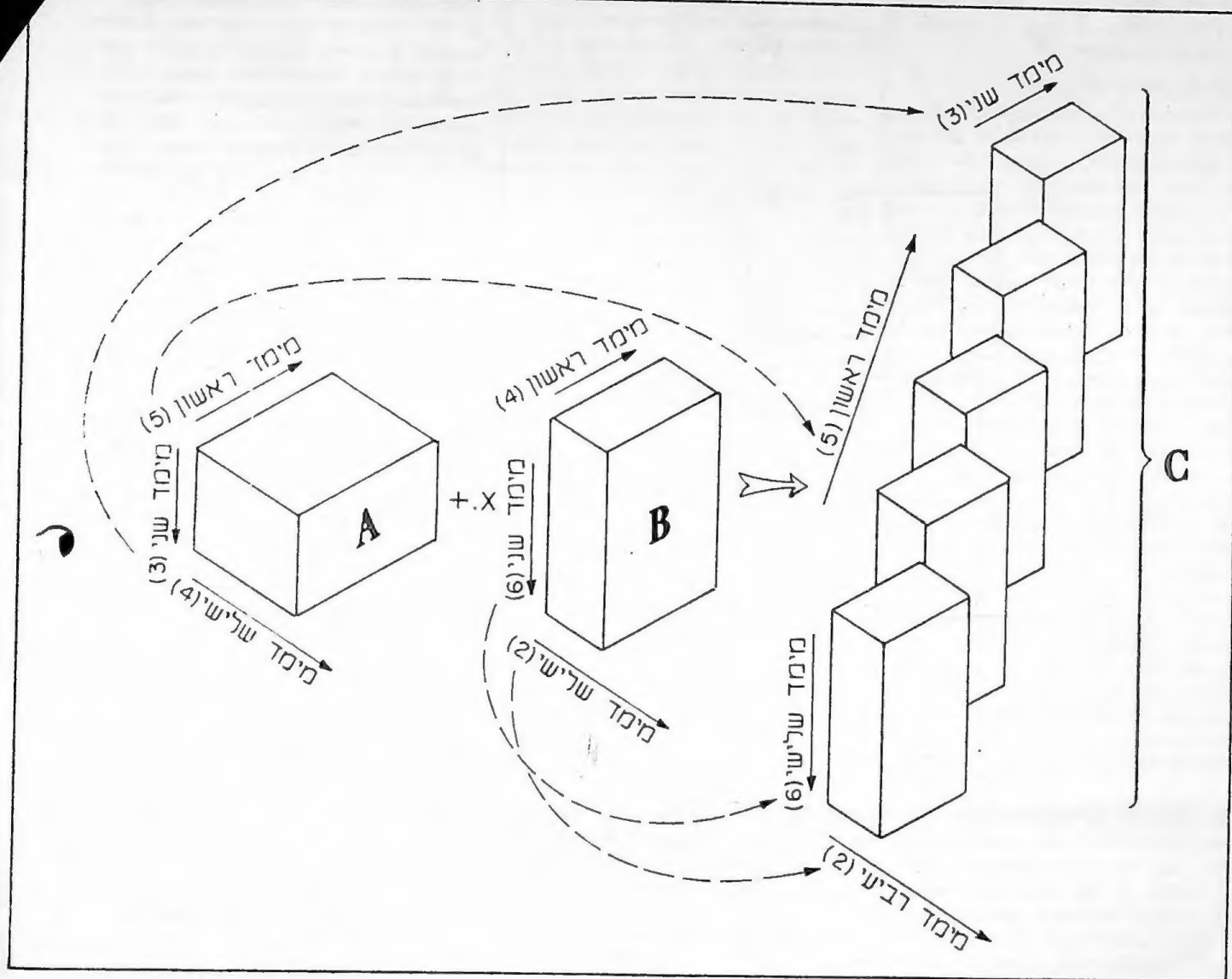
נניח ש τ זבלה תלת-ממדית, מייצג המיש-
תנה A; וכל התייחס לרכיב כלשהו של A תוך
ציון שלושת ה"אינדקסים" שלו (המהווים למעשה
קואורדינטות, כאשר כל אינדקס מתאים לאחד
הממדים של הטבלה, וסדר האינדקסים הוא מש-

```

V INVP [ ] V
V Z+INVP M;I;J;K;P;S
[1] M+M,~J+1.P+1.I+1.PM
[2] S+:[/|M
[3] L:Z+|M[1;1]X I+S
[4] K+Z1[1/2
[5] M[K,1;1.P]+M[1,K;1.P]
[6] S[K,1]+S[1,K]
[7] P[K,1]+P[1,K]
[8] P+1.P
[9] S+1.PS
[10] M[1;]+M[1;]M[1;1]
[11] M+1.P(J,1)M-(JXM[1;1]).M[1;]
[12] +LX10X I+I-1
[13] Z+M[;4P]
V

```

ציור 2: תכנית טיפוסית ב-APL.



ציור 3: כפל פנימי בין טבלות תלת-ממדיות. המיספרים בסוגריים מציינים את מיספר הרכיבים בטבלה בכיוון המימד הנדון (ראה טכסט).

של טבלת הכמויות המסופקות טבלת המח. חיתוך טבלה תלת-ממדית בגודל $10 \times 15 \times 3$ של פידיון כל מיפעל מכל לקוח עבור כל מחירון שהלקוח עשוי לבחור. כפי שרואים זאת מן המי- מדים של טבלת התוצאות, נעלם מימד המוצר אחרי ביצוע ההכפלה.

2- עמודות. החיצים המרוסקים בציור מראים את מקורם של ממדי התוצאה; כך אפשר לראות שהמימד האחרון של A והמימד הראשון של B (החייבים להיות שחים!) נעלמו. כפל פנימי של טבלות, כפי שהוגדר לעיל הוא מקרה פרטי של כפל טנזורי, פעולה שכיחה בענפי פסיקה כמו תורת-היחסות, תורת-האלאסטיות וכן במתימטיקה גבוהה; דוגמה פשוטה לכפל פני- מי של טבלה בת שלושה ממדים עם טבלה בת שני ממדים נחונה להלן:

חברה תעשייתית מסוימת מייצרת שורה של 25 מוצרים ב-10 מיפעלים השייכים לה ומספקת אותם ל-15 לקוחות קבועים. טבלה תלת-ממדית בגודל $10 \times 15 \times 25$ מתארת את הכמויות המסופקות מכל מוצר לכל לקוח מכל מיפעל.

כל לקוח יכול לבחור באחד משלושה מחירי- נים, שההבדלים ביניהם נובעים מתנאי-אשראי ומועדי אספקה; מחירונים אלה מהווים טבלה דו- ממדית (בגודל 25×3) המכילה את המחירים לכל צירוף מוצר—מיספר מחירון. כפל פנימי

למשל, כפל מטריצות³, במובן המתימטי המ- קובל, מכונה ב-APL "כפל פנימי", והוא יסומן כך:

$$A + \cdot X B \quad (18)$$

מיספר הממדים של הטבלות A ו-B איננו מוג- בל לשניים, בתנאי שהמימד האחרון של A שזה למימד הראשון של B. כך למשל ניתן לבצע ב-APL כפל פנימי בין הטבלות המיוצגות ע"י התיבות שבציור 3, כאשר אורך כל מקצוע שזה למיספר הרכיבים בכיוון מקביל לו.

התוצאה תהיה טבלה בת ארבעה ממדים, (ראה ציור מס' 3), דהיינו חמש טבלות תלת-ממדיות, שבכל אחת 3 שכבות דו-ממדיות בנוות 6 שורות

³ מטריצה— מערך של מיספרים או סמלים מתי- מטיים המסודרים בשורות ובעמודות, כך שניתן לציון כל איבר בעזרת שני אינדקסים— מראי מקומו במערך המיספרים.



פעולת APL נוספת, המהירה הרחבה של מושג כפל-המטריצות נקראת "כפל פנימי מוכלל"; היא נכתבת כמו הכפל הפנימי הרגיל, אלא שבמקום בו מופיעים סימני-הכפל והחיבור יכולים להופיע עתה סימונים אחרים, כמו $=$, \leq , \div ופעולות אלה תתבצענה במקום הכפלים והסיכומים המבוצעים בכפל פנימי, לפי הגדרתו במקובלת. הדוגמה הבאה מבחינה את המתרחש במקרה זה. נניח ש-A ו-B-1

לו רשמנו (דהיינו כפל פנימי רגיל) :

$$\begin{array}{r} 1 \ 2 \\ 3 \ 4 \ + \cdot \times \ 5 \ 3 \ 3 \\ 5 \ 6 \qquad \qquad \ 6 \ 5 \ 4 \end{array}$$

היו מתבצעות הפעולות הבאות:

$$\begin{array}{r} 1 \ 2 \times 5 \ 6 \\ 5 \ 12 \\ 5 \ + \ 12 \end{array}$$

17

1 2 × 3 5

3 10

3 + 10

13

אה הסופית :

17	13	11
39	29	25
61	45	39

ועתה, אחרי ששיכלנו את הכפל הפנימי המוכלל, מדוע לא להגדיר גם כפל חיצוני מוכלל? ואכן, "כפל" כזה קיים ב-APL והוא בא לאפשר ביצוע פעולה רצויה בין כל הזוגות האפשריים של רכיבים השייכים לשתי טבלות, בן-זוג אחד לכל טבלה. דוגמה קלאסית — לוח הכפל. למשל, לוח-הכפל של המיספרים מ-1 עד 5 איננו אלא אוסף של כל מכפלות הרכיבים של שני וקטורים, שכל אחד מהם מורכב מן המיספרים 1 ל-5.

ראשית, כזכור מנוסחה (11), הפעולה $A \rightarrow 5$

$$A \longleftrightarrow \begin{array}{cc} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{array} \qquad B \longleftrightarrow \begin{array}{ccc} 5 & 3 & 3 \\ 6 & 5 & 4 \end{array}$$

ואנן רוצים לבצע :

$$(19) \quad A + \cdot = B$$

רהיני, פעולה המתנהלת בדומה לכפל מטריצות קלאסי, אלא שבמקום לכפול את הרכיבים המתאימים אנ מבצעים השחאות ביניהם, כפי שנראה להלן.

הפעולה מתחילה בהשואה בין השורה הראשונה
נה של המטריצה השמאלית והעמודה הראשונה
של המטריצה הימנית (כזכור, התוצאה של הפ-
עולה היא 1 כאשר השתיים מתקיימות, דהיינו,
ערכו "אמת", ו-0 כאשר התוצאה היא "לא-
אמת"):

$$1 \quad 2 = 5 \quad 6$$

: וְהַתְּרַעָה :

אח"כ מתבצע סיכום: $0 + 0$

0

שלב ב' (שורה ראשונה, עמודה שניה) :

$$1 \quad 2 = 3 \quad 5$$

0 0

0 + 0

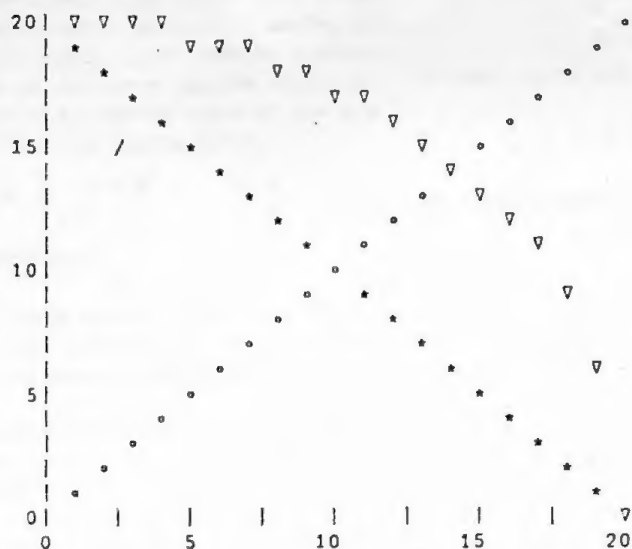
9

ובן הלאה.

ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 BCDEFGHIJKLMNOPQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 CDEFGHIJKLMNOPQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 DEFGHIJKLMNOPQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 EFGHIJKLMNOPQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 FGHIJKLMNOPQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 GHIJKLMNOPQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 HIJKLMNOPQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 IJKLMNOPQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 JKLMNOPQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 KLMNOPQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 LMNOPQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 MNOPQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 NOPQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 OPQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 PQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 QQRSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 RSTUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 STUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ
 TUVWXYZ1234567890+*+[]/.,_<=>≠∇Λ-÷÷÷*O1÷÷÷~ρew?+αf|_∇Δ°□()\\:;|TJ

ציור 4: "התחום" של APL.

30 40 PLOT X AND (20-X) AND (400-X*2)*0.5 VS X+120



ציר 5: התיאור הגראפי של הפונקציות $y = x$, $y = 20 - x$, $y = \sqrt{400 - x^2}$ כפי שבוצע על מסוף. בשפת APL.

שלו בדיוק רצוי באמצעות הטור:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

תכנית APL לחישוב N אברי הטור תראה כך:

```
EX ← 1 + X * (N) ÷ !N
```

N! נותן את החקטור $1! 2! 3! \dots N!$ X^N נותן את החקטור $X^1 X^2 X^3 \dots X^N$ אח"כ מתבצעים החילוקים ולבסוף בא הסיכום והוספת המיספר 1 המהדה את האיבר הראשון.

נניח כעת שברצוננו לכתוב טקסט מסוים עם השינוי הבא: בכל מקום שהופיעה בטקסט המ-קורי האות E צריך עתה להופיע רחח.

ראשית יש לתקתק דרך המסוף את הטקסט המקורי, שאותו נכנה בשם TEXT המאוחסן בחקטור בעל מיספר רכיבים כמיספר האותיות; כולל רחחים:

```
TEXT ← 'AN INTERESTING ARCHEOLOGICAL DISCOVERY: FORTRAN'
```

אם נכתוב כעת:

```
('E' ≠ TEXT) \ ('E' ≠ TEXT) / TEXT
```

נקבל:

```
AN INT R STING ARCH OLOGICAL  
DISCOV RY: FORTRAN
```

אנו משאירים לקורא לשכנע את עצמו שאמנם זה מה שמחקה.

לקריאה נוספת

"APL Users Manual", IBM Corporation, 1973 (GH20-0683).

"APL/360 Primer", IBM Corporation, 1973 (GH20-0689).

Iverson, K.E. 1962. "A Programming Language". Wiley, New-York.

Pakin, S. 1972. "APL/360 Reference Manual". Science Research Associates Inc, Chicago.

נספח: פעולות ודוגמאות למתקדמים

שלוש פעולות במסגרת האמור כאן הן רדוק, ציה, דחיסה ודיוח.

רדוקציה מסומנת כך: O/A

במקום המעגל יכולה לבוא פעולה כמו +, -, ×, ÷ או פעולה סקאלרית כלשהי אחרת, והר-דוקציה פירושה ביצוע הפעולה הנדונה בין כל שני רכיבים של השורות של הטבלה A.⁵

אם A מייצג את החקטור

1 2 3 אז

+ / 1 2 3

פירושו

1 + 2 + 3

דהיינו 6.

אם A מייצג את הטבלה

1 2 3 4

אזי הפעולה A / A נותנת:

1 × 2 3 × 4

דהיינו את החקטור 12.

פעולת הדחיסה באה לסלק רכיבים מחקטור:

אם נכתוב:

```
010001110111 / 'ROSTENSTREICH'
```

יפנה המחשב:

החקטור משמאל מורכב מן הספרות 1 ו-0 בלבד,

0 עבור כל רכיב של החקטור הימני שאנו רוצים

בסילוקו.

דוגמה לפעולת הדחיסה, המסומנת ב-" / " (קו

נטוי שמאלה):

נכתוב:

```
11101101111111 \ 'APLISGROOVY'
```

המחשב יפנה:

באמצעות האפסים בחקטור השמאלי הכנסנו רח-

חים בין הרכיבים של החקטור האלפביתי שמימין

לסימן הריוח.

כידוע, ייחורו של המיספר e (בסיס הלוגרית-מ

ים הטבעיים) הוא בזה שאפשר לחשב כל חזקה

נתיחה כאן לטבלות המכילות עד שני ממדים

בלבר.

ראשית, לאלה ששרדו אחרי קריאת הפרק הקודם — נדבקתם רבות; לאלה שדילגו עליו — שנתכם תידר עד שלא תחזרו ותפצחו אותו. ועתה, כמה הערות-סיכום. אקחה כי דוברי שפות-מחשב כמו פורטרן נוכחו — אחרי עיון בדוג-מות — במידת החיסכון הקיצונית במיספר הסמלים הדרוש להבעת רעיון מסוים ב-APL. לא-אחת נדרשת ב-APL פחות כתיבה מאשר לפי כללי האלגברה המקובלים. למשל, חוק מס-ההכנסה הישראלי משנת 1967 נוסח בעזרת APL בפחות מ-30 שורות. סגולה שניה של APL שהודגמה לעיל היא האפשרות לבצע פעולות פשוטות אחרי הסבר של מיספר דקות; לחכונה זו חשיבות רבה כאשר מאמצים את APL ככלי חינוכי.

תכונה שלישית היא "המקבילות". פרט לתרומה של תכונה זו לחסכוניות השפה בסמלים, יש למקבילות השלכות לצפוי מחר בעולם המחשבים. אנו עדים כבר היום לצימצום הולך וגובר בממדי המעגלים האלקטרוניים שמהם מורכבים מחשבים, צימצום המביא להגברת המהירות של ביצוע פעולות-החשבון. קיים גבול עליון תיאורטי למ-הירות זו וכבר כיום קיימים צרכים חיוניים שע-בורם גם מהירות זו איננה מספיקה. דרך אחת להתגבר על מחסום זה היא הפעלת "תזמורת" של מאות ואילפי מחשבים במקביל לביצוע משימה חיונית אחת.¹

כל-עוד היו ממדיו של מחשב בסדר-גודל של תיבת תפוחי-זהב לא בא הדבר בחשבון, אך שונה הדבר כשמגעים ממדיו לגודלה של חצי קופסת-גפרורים. שפות כמו FORTRAN ו-COBOL אינן מוכנות לעידן החישובים-במקביל; תכנית FORTRAN תצטרך לעבור ניתוח מייגע כדי לקבוע אלו מחלקיה יכולים להתבצע במקביל, מבלי שחלק אחד יצטרך לחכות לתוצאות של החלק השני.

לעומת-זאת, APL מוכנה!

פה המקום לציין גם ש-APL איננה דורשת יריעה בשפה האנגלית מאחר ואיננה משתמשת כלל במילים כמו IF, TO, GO, ELSE, השכיחות בשפות-תיכנות אחרות.

בישראל

בשורת ה-APL הגיע גם לארץ. המערכת פו-עלת באוניברסיטת בר-אילן, בטכניון, במרכז המ-דעי של חברת י.ב.מ. בחיפה וכן במשרדי חברת י.ב.מ. בחל-אביב. היישומים הם רבים ושונים והיריעה קצרה מכדי למנות את כולם. להלן אה-דים: עיבוד תוצאות ניסויים בפיסיקה (בר-אילן), הדרכת סטודנטים לביצוע ניסויים בכימיה (בר-אילן), עיבוד הצעות להתקנת מחשב (י.ב.מ.), סיוע בקביעת אבחנה רפואית (חל-השומר, י.ב.מ.). כן נכתבה בעזרת APL מערכת מידע למעקב אחר הקורה עם חיללים פצועים בביה"ח חל-השומר, באמצעות מסוף המותקן בבית-החולים והקשור בקו-טלפון למחשב של אוניברסיטת בר-אילן.

¹ כיום קיים מחשב ענק (ILLIAC IV) ובו פועלים 64 מחשבים במקביל. ראה "מדע" י"ט-1 (1974-1975), עמ' 40.